WHAT IS CLAIMED IS:

1. A controller for a synchronous DRAM comprising:

a sorting unit for receiving memory requests and sorting said memory requests based on their addresses;

a throughput maximizing unit for processing said memory requests to the synchronous DRAM in response to scheduling which maximizes the use of data slots by the synchronous DRAM.

2. A controller according to claim 1, wherein said memory requests are tagged for indicating a sending order thereof before said memory requests are sent to said sorting unit.

3. A controller according to claim 1, wherein said sorting unit tags said requests for indicating a received order thereof.

4. A controller according to claim 1, further comprising a control block for receiving a controller clock signal and developing an SDRAM clock signal by dividing said controller clock signal with a programmable divisor value.

5. A controller according to claim 4, wherein said predetermined divisor value is greater than or equal to 4 and less than or equal to 32.

6. A controller according to claim 4, wherein said throughput maximizing unit comprises a plurality of bank data paths for receiving said memory requests based on addresses sorted by said sorting unit at corresponding bank data paths.

~~7~~ 6. A controller according to claim ~~6~~ 5, wherein said control block further includes,

an arbitration unit for arbitrating said memory requests based on the previous request to the synchronous DRAM, and when conflicting memory requests are queued in one of said bank data paths, and

a constraint update unit for decoding the decisions from said arbitration unit and simultaneously updating scheduling constraints of the synchronous DRAM.

~~8~~ 7. A controller according to claim ~~7~~ 6, wherein said throughput maximizing unit further includes,

a qualification unit for qualifying said memory requests based on scheduling constraints of the synchronous DRAM, and

a command update unit for developing a command stack of said memory requests and modifying a ~~pluarlity~~ plurality of update queues which each correspond to one of said bank data paths, in response to said qualification unit.

~~9~~ 8. A controller according to claim ~~8~~ 1, further comprising a return data path for detecting the order of data returning from the synchronous DRAM with respect to said sending order of the tagged memory request.

9. 2
10. A controller according to claim 3, further comprising a return data path for detecting the order of data returning from the synchronous DRAM with respect to said received order of the tagged memory request.

Sub A2
11. A system for interfacing a processing device with a synchronous DRAM comprising:

means for developing memory requests from the processing device; and

a controller for maximizing throughput of said memory requests from the processing device to the synchronous DRAM.

12. A system according to claim 11, wherein said controller maximizes throughput based on scheduling constraints of the synchronous DRAM and arbitrates between conflicting memory requests so that data slots used by the synchronous DRAM are maximized.

Sub B4
13. A method for controlling a synchronous DRAM comprising the steps of:

(a) receiving memory requests and sorting said memory requests based on their addresses; and

(b) maximizing throughput of said memory requests to the synchronous DRAM so that use of data slots by the synchronous DRAM is maximized.

14. A method according to claim 13, further comprising the step of tagging said memory requests to indicate a sending order thereof before said memory requests are received at said step (a).

15. A method according to claim 13, further comprising the step of tagging said memory requests to indicate a received order thereof at said step (a).

16. A method according to claim 13, further comprising the steps of receiving a controller clock signal and developing an SDRAM clock signal by dividing said controller clock signal with a programmable divisor value.

17. A method according to claim 16, wherein said predetermined divisor value is greater than or equal to 4 and less than or equal to 32.

18. A method according to claim 16, wherein said step (b) receives said memory requests at a plurality of bank data paths corresponding to addresses sorted at said step (a).

19. A method according to claim 18, further comprising the steps of:

(c) arbitrating between said memory requests based on the previous request to the synchronous DRAM when conflicting memory requests are queued in one of said bank data paths; and

(d) decoding the decisions at said step (c) and simultaneously updating scheduling constraints of the synchronous DRAM.

20. A method according to claim 19, further comprising the steps of:

(e) qualifying said memory requests based on scheduling constraints of the synchronous DRAM; and

(f) developing a command stack of said memory requests in a plurality of update queues, where each of said update queues corresponds to one of said bank data paths, and modifying said update queues in response to qualifying at said step (e).

18. 11 13

21. A method according to claim 14, further comprising the step of detecting the order of data returning from the synchronous DRAM on a return data path with respect to said sending order of the tagged memory requests.

19. 12

22. A method according to claim 15, further comprising the step of detecting the order of data returning from the synchronous DRAM on a return data path with respect to said received order of the tagged memory request.

Sub A3

23. A method for interfacing a processing device with a synchronous DRAM, comprising the steps of:

(a) developing memory requests from the processing device; and

(b) maximizing throughput of said memory requests from the processing device to the synchronous DRAM.

24. A method according to claim 23, wherein said step (b) maximizes throughput based on scheduling constraints of the synchronous DRAM and arbitrates between conflicting memory requests so that data slots used by the synchronous DRAM are maximized.

23